

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2001-22638

(P2001-22638A)

(43)公開日 平成13年1月26日(2001.1.26)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード*(参考)
G 0 6 F 12/06	5 2 1	G 0 6 F 12/06	5 2 1 D 5 B 0 4 7
12/02	5 8 0	12/02	5 8 0 E 5 B 0 6 0
G 0 6 T 1/60		15/64	4 5 0 F

審査請求 未請求 請求項の数7 O L (全16頁)

(21)出願番号 特願平11-190447

(22)出願日 平成11年7月5日(1999.7.5)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 安部 雄一

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(72)発明者 中塚 康弘

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(74)代理人 100087170

弁理士 富田 和子

最終頁に続く

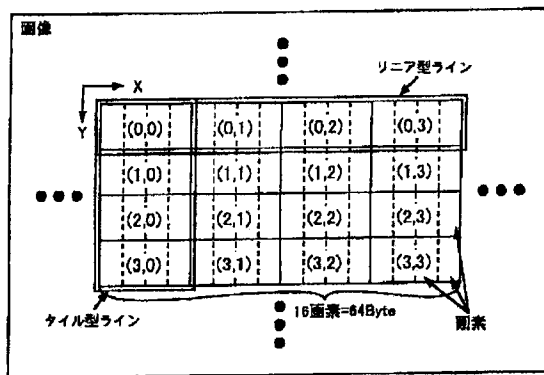
(54)【発明の名称】 情報処理システム

(57)【要約】

【課題】 メモリアクセスに関して、異なったローカリティ(局所性)を持つ処理部が混在した場合でも、それぞれのローカリティに適したメモリアクセスを可能にする情報処理システムを提供する。

【解決手段】 同時アクセス可能で、それぞれ16バイト単位で読み書き可能な、4つのモジュール0~3で構成され、64バイト単位で読み書き可能なユニファイドメモリに、4つのパックから構成される64バイトの画像データを格納する場合、リニアアクセスモード時には、4つのパック(0,0)、(0,1)(0,2)(0,3)を、それぞれ、モジュール0、1、2、3に格納する。また、タイルアクセスモード時には、4つのパック(0,0)、(1,0)、(2,0)、(3,0)を、それぞれ、モジュール0、1、2、3に格納する。

図5



【特許請求の範囲】

【請求項1】 複数のモジュールで構成されるメモリと、
当該メモリに対してアクセスを行う処理部と、
当該処理部から発行されたメモリのアドレスを、アクセスモードに従って、各モジュール毎の個別のアドレスに変換するアドレス変換部とアクセスモード及びアドレスに従って、メモリに読み書きされるデータを並び替えるデータアライナ部とを具備することを特徴とする情報処理システム。

【請求項2】 特定のサイズを有するデータ単位で読み書きすることが可能なモジュールを、N個備えたメモリと、
当該メモリとの間で、N個の前記データ単位からなるデータの読み書きを行う処理部と、
当該処理部からのアクセス要求を受けて、メモリに対してアクセスを行うメモリインタフェース部とを備え、
前記メモリインタフェース部は、
前記処理部から受け取ったN個のデータ単位のそれぞれが、異なるモジュールに格納されるように、アクセスモードに応じて、各データ単位を格納するモジュールと、
各モジュールにおける格納位置を決定することを特徴とする情報処理システム。

【請求項3】 特定のサイズを有するデータ単位で読み書きすることが可能なモジュールを、N個備えたメモリと、
当該メモリとの間で、N個の前記データ単位からなるデータの読み書きを行う処理部と、
当該処理部がメモリにアクセスする際に発行したアドレスを、アクセスモードに従って、各モジュール毎の個別アドレスにアドレス変換を行うアドレス変換部と、
処理部とメモリとの間でデータのやり取りを行う際、アクセスモードに従って、当該データを構成する前記データ単位の並び替えを行うデータアライナ部とを備えたことを特徴とする情報処理システム。

【請求項4】 前記アドレス変換部は、
N×N個のデータ単位からなる2次元配列において、同一X座標を有するデータ単位は、すべて異なるモジュールに格納され、かつ、同一Y座標を有するデータ単位は、すべて異なるモジュールに格納されるように、アドレス変換を行い、
前記データアライナ部は、
前記アドレス変換部の当該アドレス変換に応じて、データ単位の並び替えを行うことを特徴とする請求項3に記載の情報処理システム。

【請求項5】 それぞれ異なったローカリティを有する処理部と、
それぞれの処理部が共通にアクセスするユニファイドメモリと、
各処理部が使用するデータを一時的に貯めておくキャッ

シュ部と、
各処理部からのアクセス要求を受けて、ユニファイドメモリに対してメモリアクセスを行うメモリインタフェース部と、
各処理部から通知されるアクセスモードに応じて、ユニファイドメモリへアクセスするためのアドレスを変換するアドレス変換部と、
前記アクセスモードに応じて、ユニファイドメモリとやり取りするデータを並べ替えるデータアライナ部と、から構成されることを特徴とする情報処理システム。

【請求項6】 前記ユニファイドメモリは、複数のモジュールから構成され、
前記アドレス変換部は、当該各モジュール内にあることを特徴とする請求項5に記載の情報処理システム。

【請求項7】 それぞれ異なったローカリティを有する処理部と、
それぞれの処理部が共通にアクセスするユニファイドメモリと、
各処理部が使用するデータを一時的に貯めておくキャッシュ部と、
各処理部からのアクセス要求を受けて、ユニファイドメモリに対してメモリアクセスを行うメモリインタフェース部と、
各処理部から通知されるアクセスモードに応じて、ユニファイドメモリへアクセスするためのアドレスを変換するアドレス変換部と、
前記処理部と前記キャッシュ部との間に位置し、前記アクセスモードに応じて、前記処理部が読み出すデータの選択を行うデータ選択部とから構成されることを特徴とする情報処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、複数の処理部が同一のメモリにアクセスする情報処理システムに関し、特に、ユニファイドメモリアーキテクチャ（UMA）を採用したシステムにおけるメモリアクセスの高速化に関する。

【0002】

【従来の技術】情報処理システムにおける処理部は、その行う処理によって、メモリアクセスに関して、様々なローカリティ（局所性）を持つ。ここでのローカリティとは、主に空間的な局所性を意味し、複数のデータから構成されるデータ構造において、あるデータがアクセスされると、その近くに配置されたデータも近い将来アクセスされる可能性が高いという性質をいう。従来から、処理によって異なるローカリティを有効に利用するための工夫がなされてきた。

【0003】例えば、特開平8-297605号公報には、メモリ空間を小矩形であるタイルに分割し、タイル内でリニアになるようにメモリとキャッシュのアドレス

10

20

30

40

50

を管理して、CPUが画像領域にアクセスする際、タイル単位にキャッシュに転送する方式が開示されている。この方式では、テクスチャマッピングのように画像に対して2次元的なローカリティを持った処理、即ち、次のアクセスが画像の2次元的な全ての方向に対してなされる可能性がある処理では、2次元のタイルを単位にキャッシングしているためヒット率が向上する。

【0004】一方、近年、システムLSIでは、メモリシステムに、ユニファイドメモリ・アーキテクチャ（UMA）が用いられている。ユニファイドメモリ（以下、UMという）とは、従来、別々のメモリに格納されていたデータ（例えば、CPUの命令やデータと、表示画像データやテクスチャ・データ等）を、統合して格納するメモリをいう。

【0005】このようなUMAを採用した場合、UMに対して、様々な処理部からアクセスが行われることになる。つまり、異なるローカリティを有する処理部からのメモリアccessが、同じUMをアクセスする場合が生じることになる。

【0006】例えば、ビデオ入力した画像をUMに格納し、この画像をテクスチャとしてテクスチャマッピングに使用したり、或いはこの画像にフィルタを掛ける等の処理を行うシステムを考えると、これらの各処理はメモリアccessに関してそれぞれ独自のローカリティを有する。

【0007】図20は、これらの処理のローカリティを説明する図である。

【0008】同図に示すように、ビデオ入力は、画素データが左上から右下へと順に送られてくる。つまり、ビデオ入力部は、メモリアccessに関して、一次元的な（リニアな）ローカリティを有する。

【0009】これに対して、テクスチャマッピングでは、UMに格納された画素データを、はりつけ先の形状等に応じて、縦、横、斜め、とあらゆる方向にアクセスするため、メモリアccessに関して、二次元的なローカリティを有する。また、UMに格納された画像に対してフィルタリングを施すフィルタリング処理でも、一般に、注目する画素の周囲数画素を重み付け平均するため、メモリアccessに関して、二次元的なローカリティを有する。

【0010】この場合、UMに対しては、一次元的な（リニアな）ローカリティを有する処理部と二次元的なローカリティを有する処理部の両方がアクセスすることになる。

【0011】リニアなローカリティを有する処理については、アドレスをリニアに管理し、リニアなアクセスやリニアなキャッシング（バッファリング）を行えるのが望ましい。また、二次元的なローカリティを有する処理については、アドレスをタイル型に管理し、タイル型のアクセスやタイル型のキャッシングを行えるのが望まし

い。

【0012】

【発明が解決しようとする課題】前記公報記載の技術では、CPUの命令などリニアなローカリティを有したデータを格納してあるメモリ空間についてはアドレスをリニアに管理している。つまり、図21に示すように、リニアアクセス（及びリニアキャッシング）を行うか、タイルアクセス（及びタイルキャッシング）を行うかは、アクセスするアドレス領域によって決められており、同一のアドレス空間に対しリニア型アクセスとタイル型アクセスの両方を行うことはできなかった。

【0013】例えば、タイル型アドレス領域は、次のアクセスが2次元的な全ての方向になされる可能性が高いことを前提として、タイル型アクセスによってのみアクセス可能としている。この場合、テクスチャマッピングのように2次元的なローカリティを有する処理は、効率的なメモリアccessが可能で、キャッシュのヒット率の向上も期待できる。しかし、殆ど右となりの画素が次にアクセスされるビデオ入力処理についても、タイル型アドレス領域については、タイル型アクセスによって、アクセスしなければならず、リニアなローカリティを有した処理部のアクセス効率低下してしまう。

【0014】本発明の目的は、メモリアccessに関して、異なったローカリティ（局所性）を持つ処理部が混在した場合でもそれぞれのローカリティに適したメモリアccessを可能にする情報処理システムを提供することにある。

【0015】

【課題を解決するための手段】本発明に係る第1の情報処理システムは、複数のモジュールで構成されるメモリと、当該メモリに対してアクセスを行う処理部と、当該処理部から発行されたメモリのアドレスを、アクセスモードに従って、各モジュール毎の個別のアドレスに変換するアドレス変換部と、アクセスモード及びアドレスに従って、メモリに読み書きされるデータを並び替えるデータアライナ部とを具備することを特徴とする。

【0016】また、本発明に係る第2の情報処理システムは、特定のサイズを有するデータ単位で読み書きすることが可能なモジュールを、N個備えたメモリと、当該メモリとの間で、N個の前記データ単位からなるデータの読み書きを行う処理部と、当該処理部からのアクセス要求を受けて、メモリに対してアクセスを行うメモリインタフェース部とを備える。そして、前記メモリインタフェース部は、前記処理部から受け取ったN個のデータ単位のそれぞれが、異なるモジュールに格納されるように、アクセスモードに応じて、各データ単位を格納するモジュールと、各モジュールにおける格納位置を決定することを特徴とする。

【0017】また、本発明に係る第3の情報処理システムは、特定のサイズを有するデータ単位で、読み書きす

ることが可能なモジュールを、N個備えたメモリと、当該メモリとの間で、N個のデータ単位からなるデータの読み書きを行う処理部と、当該処理部がメモリにアクセスする際に発行したアドレスを、アクセスモードに従って、各モジュール毎の個別アドレスにアドレス変換を行うアドレス変換部と、処理部とメモリとの間でデータのやり取りを行う際、アクセスモードに従って、当該データを構成するデータ単位の並び替えを行うデータアライナ部とを備えることを特徴とする。

【0018】この場合において、前記アドレス変換部は、 $N \times N$ 個のデータ単位からなる2次元配列において、同一X座標を有するデータ単位は、すべて異なるモジュールに格納され、かつ、同一Y座標を有するデータ単位は、すべて異なるモジュールに格納されるように、アドレス変換を行い、前記データアライナ部は、前記アドレス変換部の当該アドレス変換に応じて、データ単位の並び替えを行うようにしてもよい。

【0019】また、本発明に係る第4の情報処理システムは、それぞれ異なったローカリティを有する処理部と、それぞれの処理部が共通にアクセスするユニファイドメモリと、各処理部が使用するデータを一時的に貯めておくキャッシュ部と、各処理部からのアクセス要求を受けて、ユニファイドメモリに対してメモリアクセスを行うメモリインタフェース部と、各処理部から通知されるアクセスモードに応じて、ユニファイドメモリへアクセスするためのアドレスを変換するアドレス変換部と、前記アクセスモードに応じて、ユニファイドメモリとやり取りするデータを並べ替えるデータアライナ部とから構成されることを特徴とする。

【0020】この場合において、前記ユニファイドメモリを複数のモジュールで構成し、前記アドレス変換部は、当該各モジュール内に設けるようにしてもよい。また、前記アドレス変換部は、前記メモリインタフェース部内に設けるようにしてもよい。

【0021】また、本発明に係る第5の情報処理システムは、それぞれ異なったローカリティを有する処理部と、それぞれの処理部が共通にアクセスするユニファイドメモリと、各処理部が使用するデータを一時的に貯めておくキャッシュ部と、各処理部からのアクセス要求を受けて、ユニファイドメモリに対してメモリアクセスを行うメモリインタフェース部と、各処理部から通知されるアクセスモードに応じて、ユニファイドメモリへアクセスするためのアドレスを変換するアドレス変換部と、前記処理部と前記キャッシュ部との間に位置し、前記アクセスモードに応じて、前記処理部が読み出すデータの選択を行うデータ選択部とから構成されることを特徴とする。

【0022】なお、本発明に係る情報処理システムは、例えば、通常の計算機システムとして、または、1チップ構成のシステムLSIとして実装される。

【0023】また、前記処理部には、例えば、CPU、ビデオ入力部、ビデオ出力部、テキストマッピング部、フィルタリング部などが該当する。

【0024】

【発明の実施の形態】以下、図面を参照しつつ、本発明の実施の形態について詳細に説明する。

【0025】図1は、本発明を適用したシステムLSIの構成を示す図である。本システムLSIは、例えば、1チップで構成される。

【0026】同図に示すように、本システムLSIは、CPU100と、ビデオ入力部110と、テキストマッピング部／フィルタリング部120と、コネクタ部101、111、121と、メモリインタフェース部130と、ユニファイドメモリ（以下、UMという）140とを備える。

【0027】CPU100は、コネクタ部101に接続され、ビデオ入力部110は、コネクタ部111に接続され、テキストマッピング部／フィルタリング部120は、コネクタ部121に接続されている。

【0028】コネクタ部101、111、121およびメモリインタフェース部130は、それぞれ、メモリバス150に接続されている。ここでは、メモリバス150のデータ幅は、512ビットとする。また、各コネクタ部101、111、121から出力されるアクセスモード選択信号が、メモリインタフェース部130に入力されている。

【0029】また、メモリインタフェース部130は、UM140にも接続されている。

【0030】CPU100、ビデオ入力部110、テキストマッピング部／フィルタリング部120は、それぞれ、異なる処理を行う処理部である。なお、テキストマッピング部とフィルタリング部は、ともに二次元的なローカリティを有しているため、代表して一つの処理部として示してある。

【0031】コネクタ部101、111、121は、各処理部とメモリバス150との間のインターフェースをとる機能ブロックである。コネクタ部101は、キャッシュ102を備え、コネクタ部111は、ライトバッファ（以下、Wバッファという）112を備え、コネクタ部121は、キャッシュ122を備える。

【0032】キャッシュ102は、CPU100が最近アクセスしたデータを保持する高速メモリである。例えば、CPU100がメモリ・リードを行う際、アクセス対象データがキャッシュ102内にあれば、そのデータがCPU100に渡される。一方、アクセス対象データがキャッシュ102内になければ、メモリバス150およびメモリインタフェース部130を介して、UM140からアクセス対象データを含む1キャッシュライン分のデータ（ここでは、512バイトのデータとする）が読み出され、アクセス対象データがCPU100に渡さ

れると共に、読み出されたキャッシュラインデータがキャッシュ102に保持される。

【0033】Wバッファ112は、ビデオ入力部110から、例えば、画素単位で入力されるデータを順次格納し、一杯になった時点で、Wバッファ112内のデータを、メモリバス150およびメモリインタフェース部130を介して、UM140に書き込む。Wバッファ112は、ビデオ入力部110とコネクタ部111との間のデータバス幅と、メモリバス150のデータバス幅との間の差を吸収し、メモリバス150の使用回数を減らすためのバッファである。つまり、ビデオ入力データをUM140に格納する場合、各画素データごとにメモリアクセスを行っていたのでは、メモリバス150の利用頻度が非常に高くなるので、複数の画素データをWバッファにためておいて、あるまとまった単位（ここでは、512バイトとする）で、UM140に書き込みを行う。

【0034】キャッシュ122は、テキストチャマッピング部／フィルタリング部120から、例えば、画素単位でのデータアクセス要求があった場合に、アクセス対象データがキャッシュ122に既に読み込まれていれば、キャッシュ122上のアクセス対象データをテキストチャマッピング部／フィルタリング部120に渡す。一方、アクセス対象データがキャッシュ122上になければ、メモリバス150およびメモリインタフェース部130を介して、UM140にアクセスを行い、アクセス対象データを含む1キャッシュライン分のデータ（ここでは、512ビットのデータとする）を読み出し、要求されたデータをテキストチャマッピング部／フィルタリング部120に渡すと共に、読みだされたキャッシュラインデータを保持する。

【0035】メモリインタフェース部130は、各処理部100、110、120からのアクセス要求を調停し、メモリアクセス要求を出している処理部の中で、実際にメモリバス150を使うことができる処理部を決定する。

【0036】調停の結果、アクセスを許可された処理部は、メモリバス150を通してメモリインタフェース部130へアドレスとアクセスモード選択信号を送出し、データの授受を行う。

【0037】メモリインタフェース部130は、受け取ったアドレス等に従って、所定のタイミングでUM140にアクセスを行い、UM140に対してデータの読み書きを行う。

【0038】メモリインターフェース部130は、アドレス変換部131とデータアライナ部132とを備える。

【0039】アドレス変換部131は、メモリバス150からメモリインタフェース部130が受け取ったアドレスを、アクセスモード選択信号に基づいて、UM140の物理アドレスへ変換する。メモリインタフェース部

130は、この物理アドレスを用いて、UM140とデータの授受を行う。

【0040】メモリインタフェース部130がUM140とデータの授受を行う際、データアライナ部132は、必要に応じて、データを所定のデータ単位で並びかえて、メモリバス150上のデータ配列とUM140上のデータ配列との間の変換を行う。

【0041】次に、UM140の構成について説明する。ここでは、UM140をDRAMを用いて構成した場合について説明する。

【0042】図2は、UM140の構成を示す図である。

【0043】同図に示すように、UM140は、 2^M 個の独立したモジュール500で構成される。例えば、出力の場合、各モジュール500からは、 2^N バイトのデータが出力され、各モジュール500からの出力データが 2^M 個分集まって、全体で、UM140から出力される $2^{(M+N)}$ バイトのデータを構成する。

【0044】また、各モジュール500は、バンクセクタ510、および、 2^L 個の独立したバンク520を備える。バンクセクタ510は、LBビットのバンクアドレス（Bアドレス）に基づいて、モジュール500の出力として、 2^L 個のバンクのうちのいずれかの出力を選択する。

【0045】また、各バンク520は、ローセクタ521と、カラムセクタ522と、センスアンプ523と、 $2^L \times 2^C$ 個のメモリセル524（1メモリセルは、 2^L バイト）とを備える。

【0046】ローセクタ521は、LRビットのローアドレス（Rアドレス）に基づいて、 2^L 個の行データ（ $2^{(L+C)}$ バイトのデータ）の中から、1つの行データを選択して、センスアンプ523に出力する。

【0047】センスアンプ523は、ローセクタ521から出力された $2^{(L+C)}$ バイトの行データを検知・増幅して、保持する。

【0048】カラムセクタ522は、LCビットのカラムアドレス（Cアドレス）に基づいて、センスアンプ523に格納されている 2^L 個のメモリセル・データの内の1つを選択し、バンク520からの出力として、 2^L バイトのデータを出力する。

【0049】なお、図2に示したUM140では、すべてのモジュール500からの出力を平行にUM140外部に出力しているが、各モジュール500からの出力を入力とするセクタを更に設け、別途供給されるモジュールアドレスに基づいて、一部のモジュールからの出力のみを、UM140の出力とするようにしてもよい。例えば、UM140に4つのモジュール0～3がある場合、1ビットのモジュールアドレス（Mアドレス）が「0」のとき、モジュール0および1の出力を出力し、1ビットのモジュールアドレスが「1」のとき、モジュ

ール2および3の出力を出力するようにしてもよい。

【0050】次に、UM140の動作について説明する。これは、一般的なマルチバンク、マルチモジュール構成のシンクロナスDRAMと同様の動作である。

【0051】UM140には、メモリインタフェース部130から、バンクアドレス、ローアドレス、カラムアドレスなどのアドレスと、リード（読み出し）、ライト（書き込み）を表わすコマンドが入力される。なお、ライトの場合は、書き込むデータも入力される。

【0052】まず、リードの際の動作について説明する。

【0053】各バンク520では、バンクアドレスによって自分が指定されると、ローアドレスに対応する $2^{(LB+LR)}$ バイトの行データが、センスアンプ523に読み出される。

【0054】センスアンプ523に読み出された行データは、カラムセクタ522に入力される。カラムセクタ522は、カラムアドレスに基づいて、センスアンプ523に読みだされた行データの中から、 2^{LC} バイトのデータの一つを選択し、バンク520から出力する。

【0055】各バンク520から出力された 2^{LC} バイトのデータは、バンクセクタ510に入力される。バンクセクタ510は、バンクアドレスに基づいて、 2^{LB} 個のバンク出力のうちから1つを選択して、モジュール出力として出力する。

【0056】前述したように、各モジュール500から出力された 2^{LB} 個の 2^{LC} バイトのデータ、計 $2^{(LB+LR)}$ バイトがUM140より出力される。UM140から読み出されたデータは、メモリインタフェース部130に渡される。

【0057】なお、センスアンプ523に行データを読み出すには、所定のサイクル数（例えば、6サイクル）が必要であるが、センスアンプ523に既に読み出されているデータをアクセスする場合は、メモリセル524から行データを読み出す必要はないので、高速に（例えば、2サイクルで）アクセスすることができる。従って、ローカリティの高いデータは、同時にセンスアンプ523に読み出されるようにすることが望ましい。

【0058】次に、ライトの際の動作について説明する。

【0059】各バンク520では、バンクアドレスで自分が指定されると、ローアドレスに対応する $2^{(LB+LR)}$ バイトの行データがセンスアンプ523に送られる。

【0060】UM140に入力された書き込みデータは、各モジュール500に入力され、バンクアドレスにより指定されたバンクのセンスアンプ123上にある行データのうち、カラムアドレスにより選択された 2^{LC} バイトのデータが書き込みデータにより書き換えられる。

【0061】ライトの場合も、リードの場合と同様に、各バンクのセンスアンプ123に既に読み出されている

データは高速に（例えば、1サイクルで）アクセスすることができるので、ローカリティの高いデータは同時にセンスアンプに読み出されるようにすることが望ましい。

【0062】以下では、UM140の構成として、LM=2、LB=4、LR=8、LC=4、LW=4の場合を考える。すなわち、UM140は、 $4 (=2^2)$ 個の独立したモジュール500で構成される。また、各モジュール500は、 $16 (=2^4)$ 個のバンク520を備え、各バンク520は、 $2^8 \times 2^4$ 個のメモリセル524を備える。また、各メモリセル524は、 2^4 バイトのデータを格納する。この場合、各モジュール500からは、それぞれ、 $16 (=2^4)$ バイトのデータが出力されるので、UM140からの出力は、 4×16 バイト = 64 バイト (= 512 ビット) となる。

【0063】次に、本実施形態で扱われる画像について説明する。

【0064】図3は、本実施形態で扱われる 512×512 画素サイズの画像の階層構造を示す図である。

【0065】画像データは、この階層的な区分に対応した形でメモリ上に格納される。実際にはこの階層とメモリ上のアドレスが対応することになり、この対応をアドレスマッピングという。

【0066】同図に示すように、本実施形態においては、 512×512 画素の画像1枚は、 8×32 のブロックから構成されるものとする。また、各ブロックは、 4×4 のセルから構成されるものとする。

【0067】そして、各セルは、 16×4 の画素から構成される。更に、各画素は、R（赤）、G（緑）、B（青）、 α （透明度）各1バイトの4成分から構成される。すなわち、1画素は、4バイト = 32 ビットのデータから構成される。従って、 512×512 画素の画像1枚は、1 Mバイトのデータで構成される。

【0068】次に、前述したような画像データをUM140に格納する際のアドレスマッピングについて説明する。

【0069】図4は、画像データをUM140に格納する際のアドレスマッピングの例を示す図である。

【0070】ここでは、UM140のうち、4 Mバイトのメモリ領域（以下、画像領域という）が、画像データの格納に使われるものとする。この場合、画像領域は、22 ビットのアドレスによって、アクセスされる。

【0071】図4の例は、この22 ビットのアドレスと、UM140における、2 ビットのモジュールアドレス (M[1:0])、4 ビットのバンクアドレス (B[3:0])、8 ビットのローアドレス (R[7:0])、4 ビットのカラムアドレス (C[3:0])、4 ビットのバイトアドレス (W[3:0]) との間のアドレスマッピングを示している。

【0072】前述したように、 512×512 画素の画像1枚は、1 Mバイトなので、先頭の2ビットは、画像

領域内でアクセスすべき画像の先頭アドレスを表わしている。この2ビットは、B[3]、B[2]として使われる。ここで、B[2]という記述は、バンクアドレスの第2ビットを表わす。ただし、Bの最下位ビットは、B[0]としている。

【0073】次の8ビットは、最上位2ビットで指定された画像内でアクセスすべきブロックの先頭アドレスを示している。ここで、上位5ビットは、画像の縦方向のアドレスYで、下位3ビットは、画像の横方向のアドレスXである。この8ビットは、ローアドレスR[7]～R[0]として使われる。

【0074】同様に、その次の4ビットは、指定されたブロック内でアクセスすべきセルの先頭アドレスを示している。ここで、上位2ビットは、縦方向のアドレスYで、下位2ビットは、横方向のアドレスXである。この4ビットは、B[1]、C[3]、B[0]、C[2]として使われる。

【0075】最後の8ビットは、指定されたセル内部のアドレスであるが、このうち上位2ビットは、セル内のライン(UM140から出力される64バイトのデータの単位)の先頭アドレスである。また、残り6ビットは、ライン内のバイトアドレスであるが、UM140に対しては、ライン単位でデータがアクセスされるので、この6ビットのライン内バイトアドレスは、UM140に入力する必要はない。

【0076】次に、図4に示したアドレスマッピング時のセル内の画素のアドレス割付について具体的に説明する。

【0077】図5は、一つのセル(16×4画素)内の画像データをメモリに格納する際の格納方式を説明する図である。

【0078】同図に示すように、セル内の各画素には、横方向(X方向)4画素のかたまり毎に、2次元のアドレスが付与されている。ここでは、第一座標をY、第二座標をXとして(Y, X)の形で記す。以下、この2次元のアドレスが付与された4画素のかたまりを、パックと呼ぶ。

【0079】パック(0, 0)～(3, 3)が、(0, 0)～(0, 3)、(1, 0)～(1, 3)、(2, 0)～(2, 3)、(3, 0)～(3, 3)の順に、UM140に格納されているとすると、図4に示したアドレス割付においては、同一のX座標を持つ4つのパックが同一モジュール(モジュールアドレス:X)に格納される。

【0080】すなわち、パック(0, 0)、(1, 0)、(2, 0)、(3, 0)がモジュール0に格納され、パック(0, 1)、(1, 1)、(2, 1)、(3, 1)がモジュール1に格納され、パック(0, 2)、(1, 2)、(2, 2)、(3, 2)がモジュール2に格納され、パック(0, 3)、(1, 3)、

(2, 3)、(3, 3)がモジュール3に格納される。

【0081】この時、同一のY座標を持つ4つのパック(例えば、パック(0, 0)、(0, 1)、(0, 2)、(0, 3))は別々のモジュール500に格納されているので、横並びの16画素に対しては同時にアクセスできる。しかし、前述したように同一のX座標を持つ4つのパック(例えば、パック(0, 0)、(1, 0)、(2, 0)、(3, 0))は同一のモジュール500に格納されているので、4×4画素に対しては同時にアクセスができない。つまり、この場合は、リニアアクセスには適しているが、タイルアクセスには適していない。

【0082】一方、パック(0, 0)～(3, 3)が、(0, 0)～(3, 0)、(0, 1)～(3, 1)、(0, 2)～(3, 2)、(0, 3)～(3, 3)の順に、UM140に格納されているとすると、図4に示したアドレス割付においては、同一のY座標を持つ4つのパックが同一モジュール(モジュールアドレス:Y)に格納される。この場合、タイルアクセスには適しているが、リニアアクセスには適していない。

【0083】リニアアクセスとタイルアクセスの両方に適したものにするためには、「同一セル内において、同一X座標を有するパックは、すべて異なるモジュールに格納されており、かつ、同一Y座標を有するパックは、すべて異なるモジュールに格納されている」必要がある。

【0084】図6は、このような条件を満たした格納方式を示す図である。同図において、縦方向(Y方向)に並んだ4つのパックは、同一モジュールに格納される。すなわち、パック(0, 0)、(1, 3)、(2, 2)、(3, 1)は、モジュール0に格納され、パック(0, 1)、(1, 0)、(2, 3)、(3, 2)は、モジュール1に格納され、パック(0, 2)、(1, 1)、(2, 0)、(3, 3)は、モジュール2に格納され、パック(0, 3)、(1, 2)、(2, 1)、(3, 0)は、モジュール3に格納される。

【0085】図6では、第0行目(Y=0)のパックは、X座標が0, 1, 2, 3と並んでいるが、第1行目(Y=1)のパックは、X座標が0, 1, 2, 3を一つずらした形、つまり、3, 0, 1, 2と並んでいる。同様に第2行目、第3行目も、さらに一つづつずらした形で並んでいる。

【0086】このような形でパックを格納すれば、「同一セル内において、同一X座標を有するパックは、すべて異なるモジュールに格納され、同一Y座標を有するパックは、すべて異なるモジュールに格納される」という条件を満たし、リニアアクセスとタイルアクセスを両立させることができる。

【0087】図7は、このような形でパックを格納する場合のアドレス割付を示す図である。

【0088】同図に示すように、図4とほぼ同様のアドレス割付になっているが、セル内のライン選択アドレスが直接的にカラムアドレスとはなっておらず、新たな2ビットのライン選択アドレスL[1], L[0]に代わっている点異なる。これはラインを選択する際にモジュール毎に異なるカラムアドレスを指定する必要があるからである。

【0089】したがって、図7に示すアドレスは、最終的にメモリセルをアクセスするまでにアドレス変換する必要がある。また、アクセスモードがリニアアクセスモードかタイルアクセスモードかによって、アドレス変換の方法が異なるため、アドレス変換時にはアクセスモード選択信号を考慮する必要がある。アドレス変換部131が、このアドレス変換を行う。

【0090】さらに、このようにずらした形で格納されたデータは、処理部でアクセスする際のデータ並びとは異なるため、処理部にデータを渡す前に、UM140から読み出したデータの並び替えを必要がある。データアライナ部132が、このデータの並び替えを行う。

【0091】次に、このアドレス変換とデータの並び替えの方法について説明する。

【0092】図8は、アクセスモードがリニアアクセスモードの場合の入力アドレス（ライン選択アドレス）に対するアドレス変換結果およびデータアライメントの対応を示す図である。

【0093】図8に示した表において、1列目はライン選択アドレスの値、2列目はモジュール番号（モジュールアドレス）を示しており、これらの組み合わせに対して、カラムアドレス（3列目）、UM140の各モジュールに格納されているパックの座標（4列目）、画像本来の画素の並びになるよう並び替えたときのパックの座標（5列目）、及び、パックの並びを正しく並び替えるための置換（6列目）を示している。

【0094】6列目にあるS1、S2の記号は、特定の置換を表す。S1は、(0, 1, 2, 3)という配列を、(1, 2, 3, 0)という配列へと変換する巡回置換を示し、S2は、(0, 1, 2, 3)という配列を(2, 3, 0, 1)という配列へと変換する巡回置換、即ち、置換S1を2度施した置換S1*S1を示す。また、1は、配列を変化させない恒等置換を示す。

【0095】図8においてアドレス変換に注目すると、カラムアドレス[C1,C0]はライン選択アドレスの値と一致している。

【0096】なお、図に示した置換は、読み出し時、すなわち、各モジュール500に格納されている状態から正しい状態に（元の画素配列のように）並べ替える際の置換である。書込みの際は、6列目の逆置換を施せばよい。1の逆置換は1、S1の逆置換はS1*S2、S2の逆置換はS2、S1*S2の逆置換は、S1である。

【0097】図9は、アクセスモードがタイルアクセス

モードの場合の入力アドレス（ライン選択アドレス）に対するアドレス変換結果およびデータアライメントの対応を示す図である。

【0098】図9に示した表の構成は、図8に示した表の構成と同じで、1列目はライン選択アドレスの値、2列目はモジュール番号を示しており、これらの組み合わせに対して、カラムアドレス（3列目）、UM140の各モジュールに格納されているパックの座標（4列目）、画像本来の画素の並びになるよう並び替えたときのパックの座標（5列目）、及び、パックの並びを正しく並び替えるための置換（6列目）を表わしている。

【0099】図9においてアドレス変換に注目すると、カラムアドレス[C1,C0]は2ビットの演算でモジュール番号からライン選択アドレスの値を引いた値になっている。

【0100】なお、図8と図9の6列目はすべて同じ置換になっており、この場合にはデータアライナ部132は、アクセスモード選択信号を必要としない。

【0101】但し、一般に、「セル内において、同一X座標を有するパックは、すべて異なるモジュールに格納されており、同一Y座標を有するパックは、すべて異なるモジュールに格納されている」という条件の格納方式をとった場合には、モードによって異なる置換が必要な場合もあり、その場合、データアライナ部132は、アクセスモード選択信号に応じて、異なる置換を行う。

【0102】次に、前述したアドレス変換を行うアドレス変換部131とUM140内の各モジュール500との間の接続形態について説明する。

【0103】図10は、メモリインタフェース部130内のアドレス変換部131とUM140内の各モジュール500との間の接続形態を示す図である。

【0104】同図に示すように、アドレス変換部131から、各モジュール500に対して、カラムアドレスの上位2ビット[C3, C2]が共通に供給される。また、カラムアドレスの下位2ビット[C1, C0]は、各モジュール500に対して個別に供給される。

【0105】メモリインタフェース部130には、アクセスを許可された処理部からメモリバス150を通して、アドレス及びアクセスモード選択信号が入力される。なお、同図では、メモリインタフェース部130に渡されるアドレスのうち、カラムアドレスの上位2ビット[C3, C2]とライン選択アドレス2ビット[L1, L0]のみを示してある。同図に示していないアドレスは、バンクアドレス及びローアドレスとして、全てのモジュール500に所定のタイミングでブロードキャストされる。

【0106】メモリインタフェース部130は、入力されたアドレスのうち、カラムアドレスの上位2ビットについては、各モジュール500にブロードキャストする。また、ライン選択アドレス2ビットとアクセスモー

ド選択信号に基づいて、図8及び図9で示したように、カラムアドレスの下位2ビットを生成する。このカラムアドレスの下位2ビットはモジュール500毎に異なるので、各モジュール500に個別に分配する。各モジュール500は、これら4ビットのカラムアドレスに従い、出力すべきデータをセンスアンプ123上から選択する。

【0107】以上説明した実施形態では、アドレス変換部131は、メモリインタフェース部130内に設けられていたが、アドレス変換部131を、各モジュール500に設けるようにしてもよい。

【0108】図11は、各モジュール500にアドレス変換部131を置いた例を示す図である。同図に示すように、各モジュール500は、アドレス変換部131を備える。また、アドレス変換部131は、モジュールアドレス・レジスタ(Mreg)1400を備えるMreg1400は、各モジュールのモジュールアドレス(モジュール番号)を格納するレジスタである。例えば、モジュール0のMreg1400には、「0」が設定され、モジュール1のMreg1400には、「1」が設定され、モジュール2のMreg1400には、「2」が設定され、モジュール3のMreg1400には、「3」が設定される。Mreg1400の値は、固定にしても可変にしてもよい。

【0109】図11の場合、メモリインタフェース部130は、メモリバス150を介して受け取るアドレスをすべてのモジュール500に所定のタイミングでブロードキャストする。

【0110】各モジュール500のアドレス変換部131は、各Mreg1400に格納されたモジュールアドレスと、メモリインタフェース部130から供給されるライン選択アドレス及びアクセスモード選択信号とに基づいて、カラムアドレスの下位2ビットを生成する。

【0111】なお、Mreg1400は、各モジュール500内のアドレス変換部131に各モジュール500のモジュールアドレス(モジュール番号)を知らせるために設けられたものである。単に、各モジュール500のモジュールアドレスを示す信号を各モジュール500のアドレス変換部131に供給するようにしてもよい。

【0112】次に、データアライナ部132の構成について説明する。

【0113】図12は、データアライナ部132の構成例を示す図である。

【0114】ここでは、簡単のため、メモリ読み出し方向のデータアライナ部132のみを示す。なお、メモリ書込み方向のデータアライナ部も、メモリ読み出し方向の場合と同様にして、巡回置換を2段重ねることで作ることができる。

【0115】図12(a)に示すように、データアライ

ナ部132は、S1部1500と、S2部1510とを備える。データアライナ部132は、ライン選択信号L0、L1に従って、図8及び図9に示したように動作する。S1部1500およびS2部1510は、それぞれ、図8、図9の6列目に示してある置換S1、S2を行うユニットである。

【0116】図12(b)に示すように、S1部1500は、セクタ1501~1504を備える。セクタ1501~1504は、選択信号L0(ライン選択アドレスL[0])の「0」、「1」に対応して、セクタの入力のうち0、1の添え字が付いているほうを選択して出力する。すなわち、S1部1500は、L0=「1」のとき、3、0、1、2の並びを0、1、2、3へ巡回置換する。

【0117】また、図12(c)に示すように、S2部1510は、セクタ1511~1514を備える。セクタ1511~1514は、選択信号L1(ライン選択アドレスL[1])の「0」、「1」に対応して、セクタの入力のうち0、1の添え字が付いているほうを選択して出力する。すなわち、S2部1510は、L1=「1」のとき、2、3、0、1の並びを、0、1、2、3へ巡回置換する。

【0118】以上のような構成を有するデータアライナ部132で適宜並び替えられた1ライン分のデータは、キャッシュ102、122等に格納される。

【0119】図13は、キャッシュの1ラインに入るパックの並びを示す図である。

【0120】図13(a)は、リニアキャッシングで、ライン選択アドレスがYのときのキャッシュの内容を表わしている。

【0121】図13(b)は、タイルキャッシングで、ライン選択アドレスがXのときのキャッシュの内容を表わしている。

【0122】次に、図6に示した方法とは異なる画像データの格納方式について説明する。

【0123】図14は、本発明の一実施形態における別の画像格納方式を表す図である。図14に示す格納方式では、同一のY座標を持つ4つのパック、即ち、横並びの16画素に対して同時にアクセスするリニアアクセスと、同一のX座標を持つ4つのパック、即ち、4×4画素に対して同時にアクセスするタイルアクセスに加えて、更に、2×2パック、即ち8×2画素の領域を同時にアクセスするモードをサポートしている。以下、このアクセスを行うモードを、8×2アクセスモードと呼ぶ。

【0124】8×2アクセスモードにおいては、例えば、パック(0、0)、(0、1)、(1、0)、(1、1)を同時にアクセスすることが可能になる。

【0125】同図において、縦方向(Y方向)に並んだ4つのパックは、同一モジュールに格納される。すなわ

ち、パック(0, 0)、(1, 2)、(2, 1)、(3, 3)は、モジュール0に格納され、パック(0, 1)、(1, 3)、(2, 0)、(3, 2)は、モジュール1に格納され、パック(0, 2)、(1, 0)、(2, 3)、(3, 1)は、モジュール2に格納され、パック(0, 3)、(1, 1)、(2, 2)、(3, 0)は、モジュール3に格納される。

【0126】図15～図17は、この場合のアドレス変換とデータの並び替えの方法を示す図である。

【0127】図15～図17に示した表の構成は、図10 8、図9に示した表の構成と同じである。

【0128】図15は、アクセスモードがリニアアクセスモードの場合を示す図である。

【0129】図16は、アクセスモードがタイルアクセスモードの場合を示す図である。

【0130】図17は、アクセスモードが8×2アクセスモードの場合を示す図である。

【0131】なお、図15～図17の置換の欄で、「0⇔2」や「2⇔3」などの記述があるが、これは、それぞれ、(0, 1, 2, 3)の中で、0と2、および、2と20 3を交換する置換、すなわち、(0, 1, 2, 3)から(2, 1, 0, 3)への置換、及び、(0, 1, 2, 3)から(0, 1, 3, 2)への置換を表わしている。

【0132】次に、本発明の別の実施形態について説明する。

【0133】図18は、本発明を適用した別のシステムLSIの構成を示す図である。

【0134】同図に示すように、本システムLSIは、データアライナ部132が、コネクタ部101、111、121に含まれている点で、図1に示したシステム30 LSIと異なる。

【0135】各処理部とコネクタ部間のデータ幅がパックのデータ幅以下の場合、必要なデータを含むパックを選択して処理部に渡せばよいので(リードの場合)、データアライナ部132は、実質的にセクタとなりデータを並べ替える処理は不要となる。したがって、この時は、データアライナ部132をメモリインタフェース部130に置くよりも小さな物量でシステムが構成できる。なお、この場合、キャッシュ102、122等には、各パックが、例えば、図8、図9、図15～図17 40の4列目に示した並びで格納されることになる。

【0136】また、更に、アドレス変換部131を各コネクタ部101、111、121に含めるようにしてもよい。この場合、各処理部がメモリインタフェース部130に送るアドレスの一部がモジュール毎に異なることになる。すなわち、各処理部からメモリインタフェース部130に対して、アドレスの一部については、モジュール毎に異なるアドレスが渡される。メモリインタフェース部130は、各処理部から渡されたアドレスのうち、モジュール毎に異なるアドレスについては、モジュ 50

ール毎に個別に送り、残りのアドレスについては、全てのモジュールにブロードキャストする。

【0137】最後に、一般のアプリケーションプログラムが動作するシステムにおける本発明によるメモリ領域の使用例について説明する。

【0138】図19は、本発明を適用したUM140のメモリ領域の使用例を示す図である。

【0139】この場合、UM140を、CPU100上で動作しているアプリケーションが直接アクセスする領域1900と、表示画像やテキストなどを格納しておく画像領域1910とに分けている。そして、一般のアプリケーションが画像をテキストチャとして登録したり、ビデオ入力を行う際には、必ず標準のライブラリ(関数の集まり)を使用して、これらの処理を行うようにし、これらのライブラリのドライバ(ライブラリ関数の実体)に対してのみ、画像領域1910へのアクセスを許可しておく。この場合、ドライバは、画像領域1910にアクセスする際には、図6や図14に示したようなリニアアクセスやタイルアクセスが可能な格納方式に即してアクセスする。

【0140】このようにしておけば、新しいシステムを提供する際にはライブラリのドライバを共に提供することで、アプリケーションプログラムやコンパイラを変更することなく、画像領域1910において異なるアクセス方法(例えば、リニアアクセスとタイルアクセス)を両立させることができる。

【0141】画像以外に音声などを扱う場合にも、一般のアプリケーションプログラムが動作するシステムにおいては、CPU上で動作しているアプリケーションがアクセスする領域と、画像や音声などCPU以外のリソースがアクセスする領域とを分けておくことで、CPU以外のリソースがアクセスする領域においてアプリケーションプログラムやコンパイラを変更することなく、ある特定の領域においてリニアアクセス(リニアキャッシング)とタイルアクセス(タイルキャッシング)を両立させることができる。

【0142】

【発明の効果】以上詳細に説明したように、本発明によれば、同一のアドレス空間に対して、リニアアクセスとタイルアクセス等、異なるアクセス方法でアクセスすることが可能となり、これによって、メモリアクセスに関して、異なったローカリティ(局所性)を持つ処理部が混在した場合でも、それぞれのローカリティに適したメモリアクセスが可能になる。

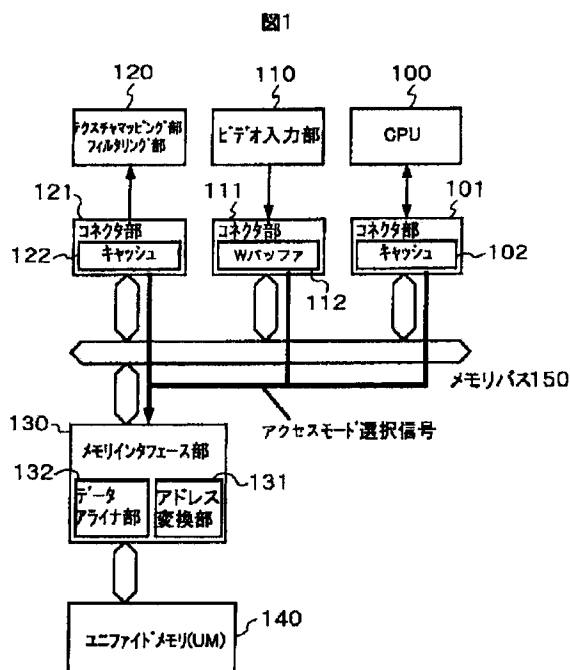
【0143】その結果、異なったローカリティ(局所性)を持つ処理部が混在した場合でも、メモリへのアクセス効率の低下を防止できる。また、各処理部がキャッシュを備えている場合は、ヒット率の向上が期待でき、処理速度の向上が図れる。

【図面の簡単な説明】

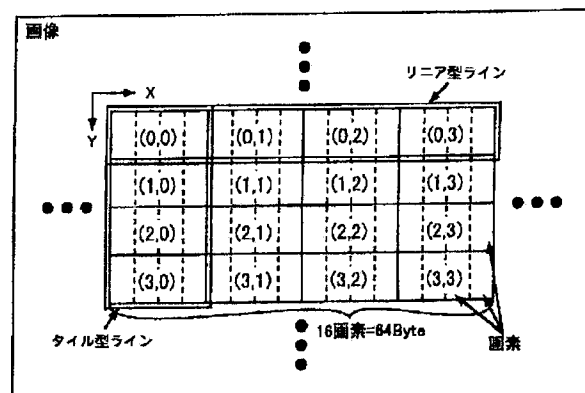
- 【図1】 本発明によるシステムLSIのブロック図である。
- 【図2】 ユニファイドメモリの構成を示すブロック図である。
- 【図3】 画像の階層構造を説明する図である。
- 【図4】 画像データをメモリに格納する際のアドレスマッピングの例を示す図である。
- 【図5】 画像をメモリに格納する際の格納方式の例を示す図である。
- 【図6】 本発明による画像格納方式を説明する図である。
- 【図7】 本発明による画像格納方式で画像データを格納する際のアドレスマッピングを示す図である。
- 【図8】 リニアアクセスモード時の入力アドレスに対するアドレス変換結果およびデータアライメントの対応を示す図である。
- 【図9】 タイルアクセスモード時の入力アドレスに対するアドレス変換結果およびデータアライメントの対応を示す図である。
- 【図10】 メモリインタフェース部130と各モジュール500との間の接続形態を示す図である。
- 【図11】 各モジュール500にアドレス変換部131を置いた例を示す図である。
- 【図12】 データアライナ部の構成を示すブロック図である。
- 【図13】 キャッシュ内のパックの配置を示す図である。

- * 【図14】 本発明による別の画像格納方式を説明する図である。
- 【図15】 リニアアクセスモード時の入力アドレスに対するアドレス変換結果およびデータアライメントの対応を示す図である。
- 【図16】 タイルアクセスモード時の入力アドレスに対するアドレス変換結果およびデータアライメントの対応を示す図である。
- 【図17】 8×2アクセスモードの時の入力アドレスに対するアドレス変換結果およびデータアライメントの対応を示す図である。
- 【図18】 本発明による別のシステムLSIのブロック図である。
- 【図19】 一般のアプリケーションプログラムが動作するシステムにおけるUMの使用例を示す図である。
- 【図20】 ローカリティの概念を説明する図である。
- 【図21】 従来方式によるメモリアクセスの概要を説明する図である。
- 【符号の説明】
- 100 CPU
- 110 ビデオ入力部
- 120 テクスチャマッピング部／フィルタリング部
- 101, 111, 121 コネクタ部
- 130 メモリインタフェース部
- 131 アドレス変換部
- 132 データアライナ部
- 140 ユニファイドメモリ (UM)

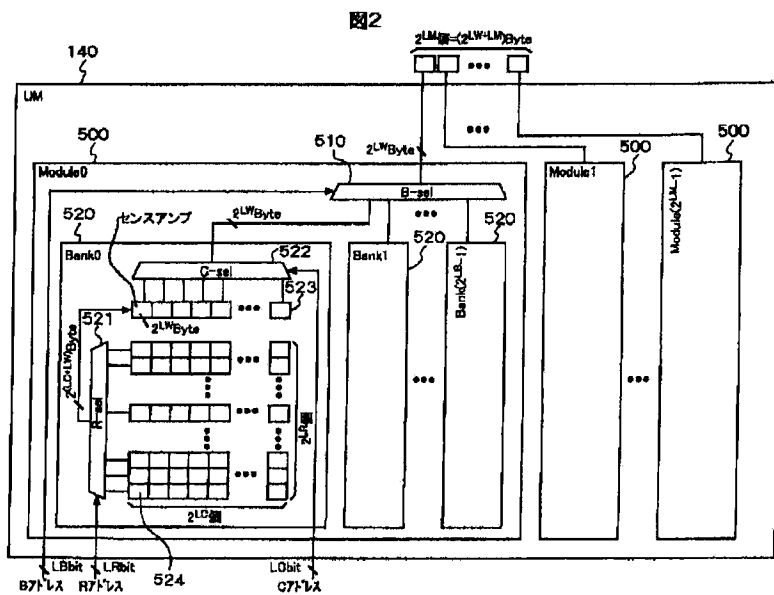
【図1】



【図5】



【図2】



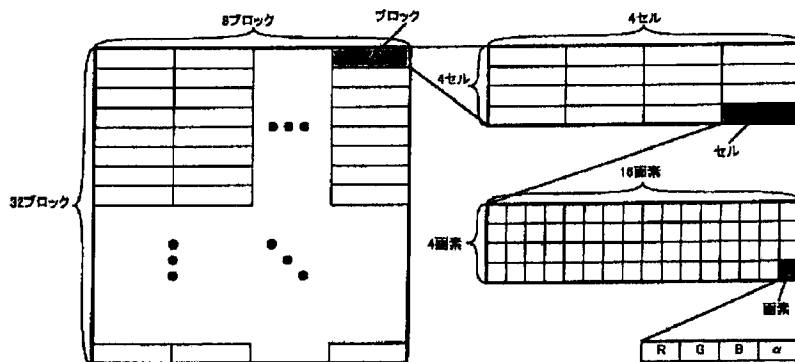
【図8】

図8

ライン 選択 信号	モジュール	C1, C0	格納 座標	ライン位 の座標	置換
0 (2'b00)	0	0	(0,0)	(0,0)	1
	1	0	(0,1)	(0,1)	
	2	0	(0,2)	(0,2)	
	3	0	(0,3)	(0,3)	
1 (2'b01)	0	1	(1,3)	(1,0)	S1
	1	1	(1,0)	(1,1)	
	2	1	(1,1)	(1,2)	
	3	1	(1,2)	(1,3)	
2 (2'b10)	0	2	(2,2)	(2,0)	S2
	1	2	(2,3)	(2,1)	
	2	2	(2,0)	(2,2)	
	3	2	(2,1)	(2,3)	
3 (2'b11)	0	3	(3,1)	(3,0)	S1+S2
	1	3	(3,2)	(3,1)	
	2	3	(3,3)	(3,2)	
	3	3	(3,0)	(3,3)	

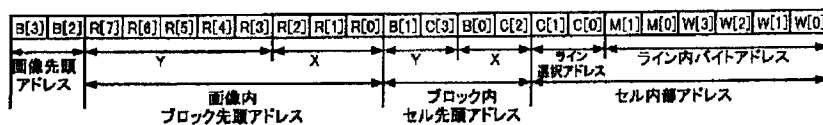
【図3】

図3



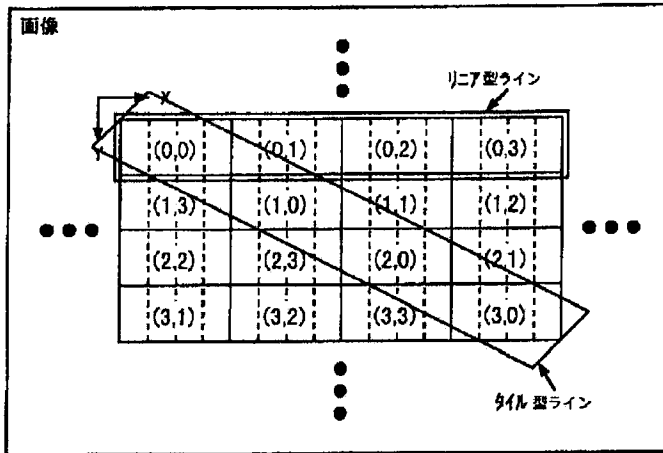
【図4】

図4



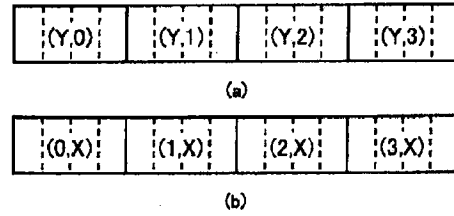
【図6】

図6



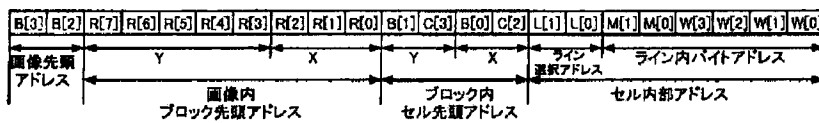
【図13】

図13



【図7】

図7



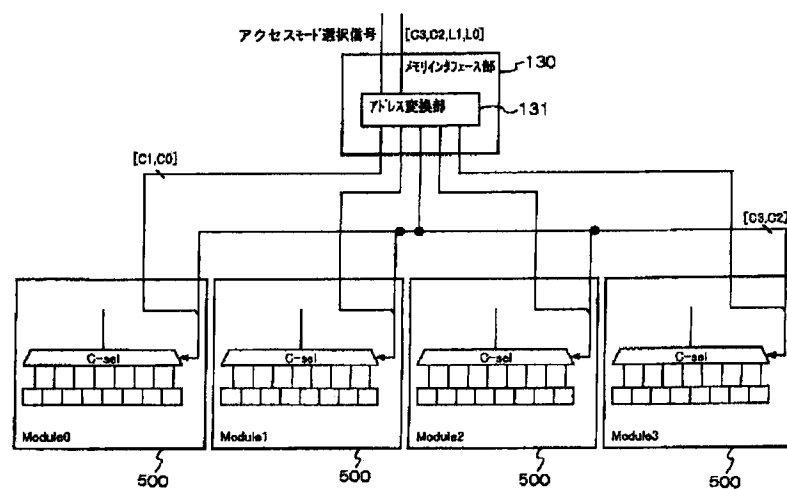
【図9】

図9

ライン 選択 番号	モジュール	C1, C0	格納 座標	ライン後 の座標	置換
0 (2'b00)	0	0	(0,0)	(0,0)	1
	1	1	(1,0)	(1,0)	
	2	2	(2,0)	(2,0)	
	3	3	(3,0)	(3,0)	
1 (2'b01)	0	3	(3,1)	(0,1)	S1
	1	0	(0,1)	(1,1)	
	2	1	(1,1)	(2,1)	
	3	2	(2,1)	(3,1)	
2 (2'b10)	0	2	(2,2)	(0,2)	S2
	1	3	(3,2)	(1,2)	
	2	0	(0,2)	(2,2)	
	3	1	(1,2)	(3,2)	
3 (2'b11)	0	1	(1,3)	(0,3)	S1*S2
	1	2	(2,3)	(1,3)	
	2	3	(3,3)	(2,3)	
	3	0	(0,3)	(3,3)	

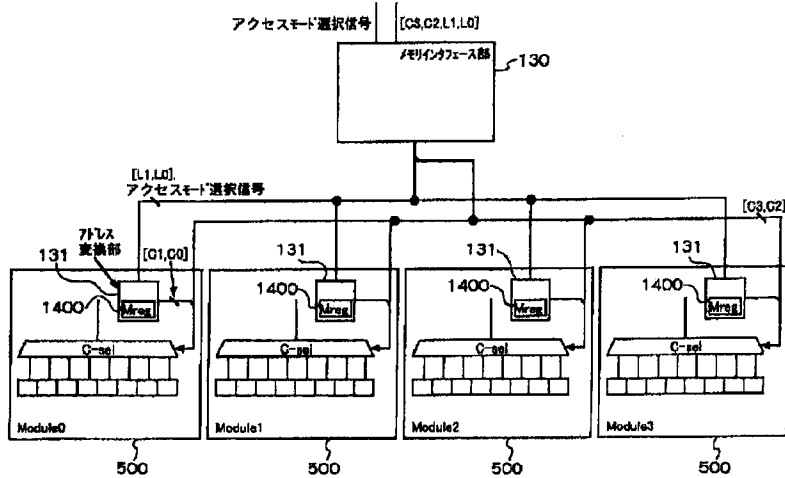
【図10】

図10



【図11】

図11



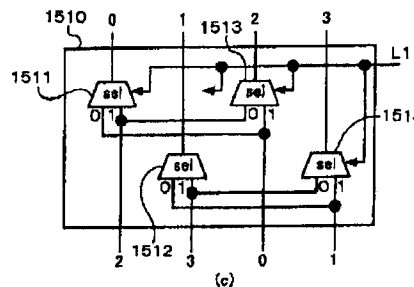
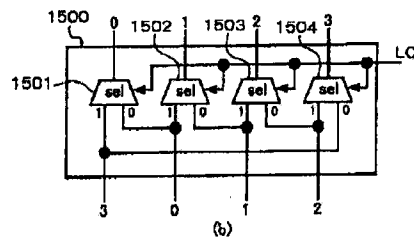
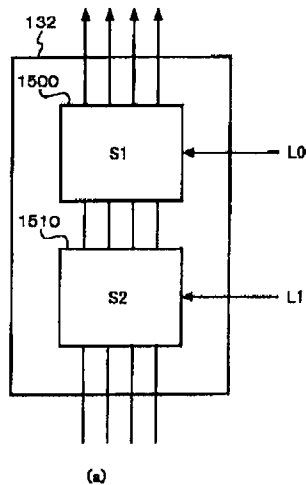
【図15】

図15

ライン 選択 信号	モジュール	C1, C0	格納 座標	ライン後 の座標	置換
0 (2'b00)	0	0	(0,0)	(0,0)	1
	1	0	(0,1)	(0,1)	
	2	0	(0,2)	(0,2)	
	3	0	(0,3)	(0,3)	
1 (2'b01)	0	1	(1,2)	(1,0)	0⇔2 *
	1	1	(1,3)	(1,1)	
	2	1	(1,0)	(1,2)	
	3	1	(1,1)	(1,3)	
2 (2'b10)	0	2	(2,1)	(2,0)	0⇔1 *
	1	2	(2,0)	(2,1)	
	2	2	(2,3)	(2,2)	
	3	2	(2,2)	(2,3)	
3 (2'b11)	0	3	(3,3)	(3,0)	0⇔3 *
	1	3	(3,2)	(3,1)	
	2	3	(3,1)	(3,2)	
	3	3	(3,0)	(3,3)	

【図12】

図12



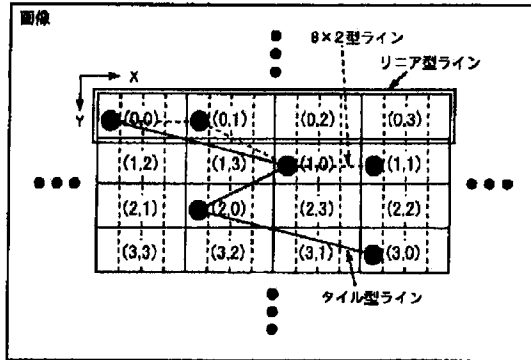
【図16】

図16

ライン 選択 信号	モジュール	C1, C0	格納 座標	ライン後 の座標	置換
0 (2'b00)	0	0	(0,0)	(0,0)	1⇔2
	1	2	(2,0)	(1,0)	
	2	1	(1,0)	(2,0)	
	3	3	(3,0)	(3,0)	
1 (2'b01)	0	2	(2,1)	(0,1)	S2 *
	1	0	(0,1)	(1,1)	
	2	3	(3,1)	(2,1)	
	3	1	(1,1)	(3,1)	
2 (2'b10)	0	1	(1,2)	(0,2)	S2 *
	1	3	(3,2)	(1,2)	
	2	0	(0,2)	(2,2)	
	3	2	(2,2)	(3,2)	
3 (2'b11)	0	3	(3,3)	(0,3)	0⇔3
	1	1	(1,3)	(1,3)	
	2	2	(2,3)	(2,3)	
	3	0	(0,3)	(3,3)	

【図14】

図14



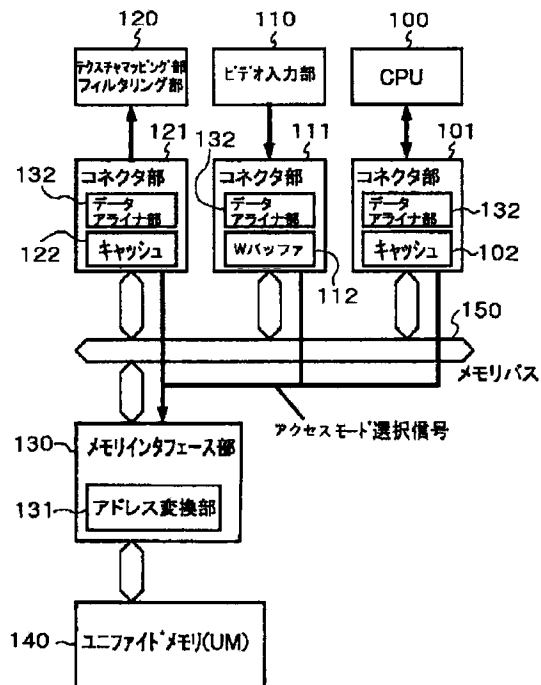
【図17】

図17

ライン 選択 信号	モジュール C1, C0	格納 座標	ライン後 の座標	置換
0 (2'b00)	0	0	(0,0)	1
	1	0	(0,1)	
	2	1	(1,0)	
	3	1	(1,1)	
1 (2'b01)	0	1	(1,2)	0⇔2 *
	1	1	(1,3)	
	2	0	(0,2)	
	3	0	(0,3)	
2 (2'b10)	0	2	(2,1)	0⇔1 *
	1	2	(2,0)	
	2	3	(3,1)	
	3	3	(3,0)	
3 (2'b11)	0	3	(3,3)	0⇔3 *
	1	3	(3,2)	
	2	2	(2,3)	
	3	2	(2,2)	

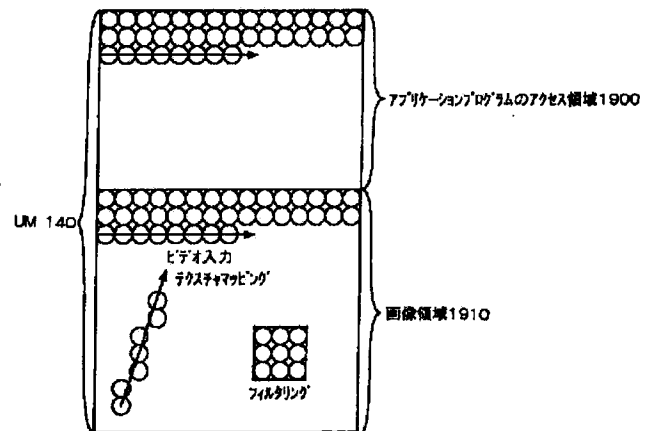
【図18】

図18



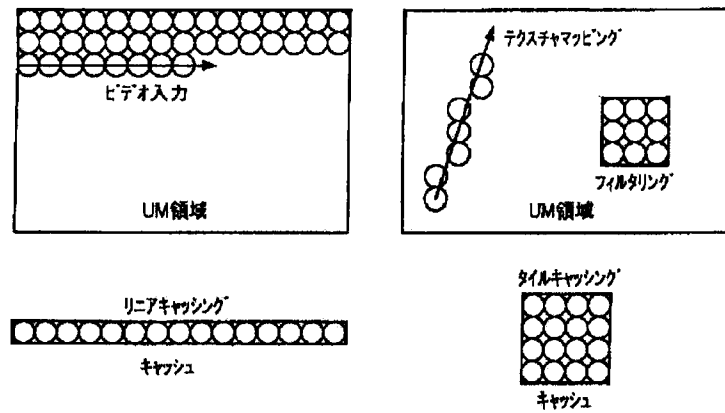
【図19】

図19



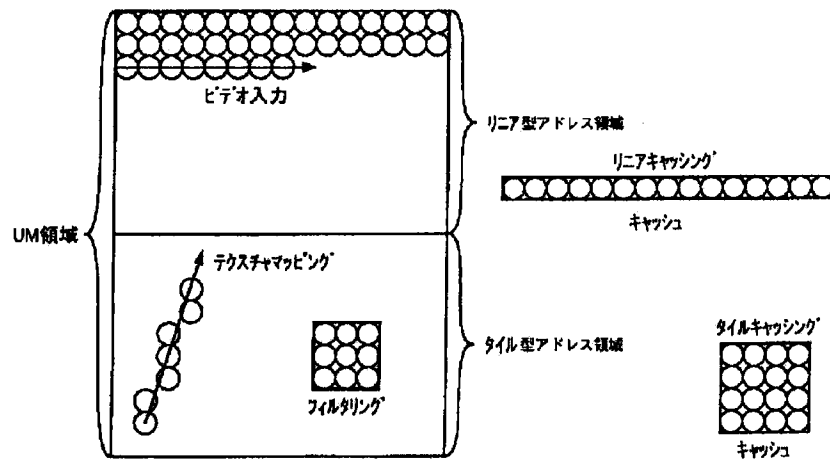
【図20】

図20



【図21】

図21



フロントページの続き

(72)発明者 松尾 茂
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内
(72)発明者 下村 哲也
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内

(72)発明者 城 学
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内
(72)発明者 佐藤 潤
東京都小平市上水本町五丁目20番1号 株
式会社日立製作所半導体グループ内

F ターム(参考) 5B047 AB04 EA09 EB01 EB06 EB13
5B060 AA13 AC13 GA06 GA11 GA16
MM13

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-022638

(43)Date of publication of application : 26.01.2001

(51)Int.Cl. G06F 12/06

G06F 12/02

G06T 1/60

(21)Application number : 11-190447

(71)Applicant : HITACHI LTD

(22)Date of filing : 05.07.1999

(72)Inventor : ABE YUICHI
NAKATSUKA YASUHIRO
MATSUO SHIGERU
SHIMOMURA TETSUYA
JO MANABU
SATO JUN

(54) INFORMATION PROCESSING SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an information processing system enabling access suitable for each of localities even when processing parts having different localities are mixed concerning memory access.

SOLUTION: This information processing system is composed of four modules 0 to 3 accessible at the same time and each capable of reading/writing for the unit of 16 bytes and when storing the image data of 64 bytes composed of four packs in a unified memory capable of reading/writing for the unit of 64 bytes, in a linear access mode, four packs (0, 0), (0, 1), (0, 2) and (0, 3) are stored in the modules 0, 1, 2 and 3, respectively. At the time of tile access mode, four packs (0, 0), (1, 0), (2, 0) and (3, 0) are stored in the modules 0, 1, 2 and 3, respectively.

